ISSN 1561-5359. Штучний інтелект, 2018, № 2

УДК 004.89

А.В. Карканица[1], В.В. Краснопрошин[2]
[1]Гродненский государственный университет имени Янки Купалы, Беларусь
ул. Ожешко, 22, г. Гродно, 230023
[2]Белорусский государственный университет, Беларусь
пр. Независимости, 4, г. Минск, 220030

МОДЕЛИРОВАНИЕ ПРЕДМЕТНЫХ ОБЛАСТЕЙ ДЛЯ АДАПТИВНЫХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

H.V. Karkanitsa[1], V.V. Krasnoproshin[2]
[1]Yanka Kupala State University of Grodno, Belarus
22, Ozheshko st., Grodno, 230023
[2]Belarusian State University, Belarus
4, Nezavisimosti av., Minsk, 220030

SUBJECT DOMAINS MODELING FOR ADAPTIVE DECISION SUPPORT SYSTEMS

Рассматривается проблема моделирования предметной области (ПрО) для адаптивных систем поддержки принятия решений (АСППР), функционирующих в условиях быстро изменяющейся среды. Предлагается методология построения адаптивной модели ПрО, основанная на ее теоретико-множественном представлении и трехэтапной схеме моделирования. Практическая значимость результатов заключается в возможности использовать методологию для построения модели ПрО АСППР, устранить статичность модели и определить механизмы ее адаптации к изменениям среды.

Ключевые слова: моделирование предметной области**,** адаптивная модель, адаптивная СППР

The problem of subject domain (SD) modeling for the construction of Adaptive Desicion Support Systems (ADSS) that function in a changing decision environment is considered. A methodology for constructing an adaptive SD-model based on its set-theoretic representation and the three-stage modeling scheme is proposed. The practical significance of the research consists in the possibility of using the methodology to construct a SD-model for ADSS, to eliminate the static nature of the model and to determine the mechanisms of its adaptation to environmental changes.

Keywords: subject domain modeling, adaptive model, adaptive DSS

Введение

Любая задача принятия решений (ЗПР) решается в условиях некоторой среды: информационной, социальной, экономической, политической, образовательной, природной или техногенной. Среда принятия решений в настоящее время быстро меняется [1]. С одной стороны, существенно возрастает интенсивность появления новых знаний, хранение которых приобретает распределенный характер. А с другой стороны, сетевой контент и знания из некоторых предметных областей быстро устаревают [2,3].

В таких условиях повышаются риски использования недостоверных, не обоснованных, а также устаревших знаний. Одним из способов устранения указанных рисков является привлечение квалифицированных экспертов. Опыт и актуальность знаний экспертов позволяет обеспечить не только качество, но и сократить время решения задачи за счет декомпозиции исходной задачи на подзадачи. На начальном этапе жизненного цикла ЗПР количество источников знаний о предметной области (ПрО) неизвестно. В процессе решения ЗПР (в силу изменчивости среды) требуется постоянное уточнение ПрО. Для своевременной актуализации знаний необходимо иметь механизмы, с помощью которых эксперты, независимо от места их нахождения, могли бы модифицировать соответствующие фрагменты ПрО.

В такой ситуации актуальной проблемой является создание адаптивных систем поддержки принятия решений (АСППР), способных реагировать на изменение

условий принятия решений. Разработка АСППР требует построения адаптивной модели ПрО и соответствующих алгоритмов для ее модификации. В данном исследовании представлена методология построения адаптивной модели ПрО под-класса ЗПР, которые решаются на основе знаний распределенных экспертов.

Постановка задачи

Для формализации постановки задачи используем онтологический подход. Пусть имеется некоторая сложно структурированная задача *Task*, включающая постановку общей задачи S_0 и ее *n* атомарных подзадач:

$$Task = (S_0, S^1, S^2, \ldots, S^n), n \to \infty. \quad (1)$$

В рамках каждой S^i известна постановки задачи. В сцене решения *Scene* участвуют центр *Center* и *k* территориально распределенных экспертов E^i, являющихся источниками компетентных знаний:

$$Scene = (S_0, Center, E^1, E^2, \ldots, E^k), k \le n. \quad (2)$$

Общую схему решения задачи (1) содержательно можно описать следующим образом. Центр выполняет предварительную декомпозицию исходной задачи на подзадачи S^i и распределяет их между экспертами (источниками знаний). Эксперты известны центру, компетентны, между задачами S^i и экспертами E^i установлено взаимно однозначное соответствие. В процессе решения любой эксперт (ввиду отсутствия актуальных знаний или с целью уменьшения сложности текущей задачи) может принять решение о дальнейшей декомпозиции задачи S^i на атомарные подзадачи. Для решения атомарных подзадач необходимы знания Z_i. Знания могут быть представлены в различных форматах и в сумме составляют ПрО. Задача (1) решается, как правило, в условиях неконтролируемых изменений внешней среды *Env*. То есть, возможны изменения состава кортежей (1), (2):

- изменение множества подзадач исходной задачи;
- изменение целей, требований, ограничений подзадач;
- количественное или качественное изменение состава группы экспертов;
- устаревание знаний и появление новейших знаний.

Требуется разработать методологию построения адаптивной модели ПрО задачи (1). Под адаптивной моделью будем понимать модель, которая устраняет статичность объектов ПрО и удовлетворяет следующим требованиям:

- отражает структурную и информационную динамику ПрО на всем временном интервале: от постановки задачи до получения решения;
- имеет механизмы адаптации к изменениям среды, как с точки зрения ее структуры, так и с точки зрения актуальности и полноты знаний.

Таким образом, на этапе моделирования ПрО, в соответствии с отмеченными требованиями, необходимо решить две основные проблемы:

- формального представления задачи, отражающего ее структуру (структурная составляющая ПрО), и знаний, необходимых для ее решения (информационная составляющая ПрО);
- задания механизмов модификации как структурной, так и информационной составляющих модели.

Анализ методологий

В программной инженерии (как и в других научных дисциплинах) моделирование ПрО реализуется, как правило, в три этапа, на каждом из которых строятся соответственно концептуальная, онтологическая и физическая модели ПрО.

В настоящее время спектр методов моделирования, применяемых на каждом из обозначенных выше этапов, достаточно широк: от простейших графических нотаций и строгих математических структур до объектно-ориентированных языков моделирования. Вместе с тем все существующие на сегодняшний момент методологии можно разделить на два основных класса: структурные и объектно-ориентированные.

В основе структурных методов лежит методология структурного анализа и проектирования SADT, предложенная Д.Россом и развитая впоследствии П.Ченом, П.Коддом и Т.ДеМарко. Развитие данной методологии привело к появлению семейства стандартов

моделирования IDEF [4]. Основополагающей в этих стандартах является модель П.Чена «сущность-связь» или Entity Relationship Model (ERM), которая описывет значимые сущности ПрО и связи между ними [5]. Связи представляются в форме иерархии, известной как ERD-диаграммы. Фактически модель «сущность-связь» и является концептуальной моделью (КМ) и, в общем случае, в нотации П.Чена может быть представлена следующим образом:

$$modC = (Entity, Relation, ERD), \text{где} \quad (3)$$

Entity – конечное множество сущностей ПрО; *Relation* – множество типов связей.

В контексте задачи (1) значимыми сущностями являются задачи *Task*, эксперты *Expert* и знания *Z*. Компонент *Relation* позволяет определить три типа связей «задача-подзадача», «задача-эксперт», «задача-знания». ERD-модель представляется ациклическим графом, узлы которого определяют иерархию подзадач, а дуги – уровень их вложенности:

$$Entity = (Task, E, Z) \quad (4)$$

$$Relation = \begin{cases} f_1(Task_i, Task_j) \\ f_2(E, Task) \\ f_3(Task, Z) \end{cases} \quad (5)$$

$$modC = (Entity, Relation, ERD) \quad (6)$$

Таким образом, можно сделать вывод о том, что КМ (4)-(6) покрывает важные свойства ПрО ЗПР, моделируя ее основные сущности. Однако, вполне очевидны и противоречия, которые возникают между свойствами ER-модели и требованием ее адаптивности. В частности:

- конечность множества сущностей противоречит требованию, что количество подзадач целевой задачи и источников знаний для их решения заранее неизвестно;
- алгоритм построения модели позволяет получить только одномоментный статический «снимок» ПрО в виде жестко формализованной ER-модели, что не позволяет манипулировать количеством экземпляров сущностей *Expert* и *Task*;
- средства алгоритма построения модели ориентированы на локальные источники знаний, что осложняет их использование территориально распределенными экспертами;
- в модели (5) отсутствуют составляющие, описывающие модель представления знаний *Z* экспертов.

Таким образом, методология П.Чена позволяет построить статическую ER-модель ПрО. На практике такая модель строится, как правило, бизнес-аналитиками единожды. Любое изменение в модели ПрО требует вмешательства разработчиков, пересборки всех компонентов ПО и их развертывания на рабочих местах пользователей. Поэтому прямое использование методологии «сущность-связь» и соответствующей нотации (например, IDEF1X) для исследуемого класса задач невозможно.

Вместе с тем, отмеченные выше свойства модели «сущность-связь» позволяют рассматривать методологию П.Чена в качестве основы для решения задачи первого уровня концептуализации ПрО: выделить значимые сущности, их атрибуты и отношения. На следующем этапе моделирования потребуется решить проблему «конечности» множества сущностей и статичности их структуры.

Этап построения онтологической модели (ОМ) может быть успешно реали-зован с использованием объектно-ориентированных методологий и соответствующих им CASE-средств. Структурообразующим элементом модели в объектно-ориентированных методологиях является объект [6]. В роли объектов выступают конкретные сущности ПрО, например, клиент, заказ, услуга. Каждый объект характеризуется своим состоянием – набором атрибутов, значения которых определяют конкретное состояние объекта, а также набором операций для его изменения. Модель ПрО описывается средствами языка моделирования (например, UML).

Анализируя возможность использования объектно-ориентированных методологий для построения ОМ ПрО (для исследуемого класса ЗПР) следует заметить, что:

- язык моделирования представляет собой

только нотацию, но не методологию моделирования;
- использование таких методологий требует привлечения квалифицированных кадров;
- моделируются хорошо структурированные предметные области с постоянными источниками знаний;
- моделирование возможно внутри локальной группы участников проекта.

Таким образом, свойства объектно-ориентированных методологий также не в полной мере соответствует свойствам рассматриваемого подкласса ЗПР. Эти методологии в большей степени ориентированы на решение задач со «статическими» предметными областями, модели которых не изменяются в процессе эксплуатации программной системы. Онтологии таких ПрО представляются в виде конечного множества сущностей, типы и структура которых известны еще на этапе проектирования.

Задачи с динамическим характером ПрО, где изменение структуры и параметров модели ПрО является естественной реакцией на изменения внешней среды, требуют принципиально другого подхода к моделированию. Предлагается, следуя традиционной схеме трехэтапного моделирования ПрО (КМ→ОМ→ФМ), разработать методологию, которая на каждом из выделенных этапов позволит устранить статичность модели и определить механизмы ее адаптации к изменениям среды.

Теоретико-множественный подход к моделированию

Одним из возможных вариантов устранения «статичности» структурной и информационной составляющих ПрО является построение модели более высокого уровня абстракции. Например, такую модель можно строить в виде параметризированного шаблона, на основании которого тиражируются сущности ПрО и уточняется их структура за счет изменения значений параметров шаблона.

Предлагается реализация этого подхода на основе теоретико-множественно-го представления модели ПрО [7]. Согласно такому представлению, любую модель формально можно описать с помощью множества объектов $O_1, O_2, \dots O_n$, связанных между собой семантическими отношениями. Тогда в самом общем виде онтологическую модель (ОМ) ПрО можно представить в виде следующего кортежа:

$$modO = \{O_t, C, M, R, R', t) , \qquad (7)$$

где O_t – множетво объектов предметной области; C – множество классов объектов; M – множество методов (алгоритмов, функций и процедур модификации ПрО); R – множество семантических отношений между объектами; R' – множество ограничений целостности модели; t – временной параметр.

Основным элементом модели (7) является объект, причем допускается их неограниченное количество. Под объектом понимается некоторая сущность, имеющая фиксированное множество атрибутов, состояние и поведение. Состояние определяется атрибутами и семантическими отношениями, поведение – методами. Объекты модели могут быть классифицированы, то есть, отнесены к некоторым классам. В соответствии с таким представлением модель ПрО можно представить множеством экземпляров этих классов.

Для исследуемого типа ЗПР такими классами являются: *Task* – задача, *Center* – центр, инициирующий задачу, *Expert* – эксперт, *Env* – окружение (среда). Ограничением целостности модели является единственность экземпляров классов *Center* и *Env*. Количество экземпляров классов *Task* и *Expert* непостоянно и может изменяться во времени.

Необходимо отметить, что уже на начальном этапе построения модели теоретико-множественное представление позволяет выделить элементы модели и задать ограничения, соответствующие свойствам ПрО задачи. Использование понятия классов объектов позволяет многократно тиражировать объекты, обладающие заданными характеристиками. Тем самым стираются границы модели предметной области ЗПР: исходная задача *Task* может быть декомпозирована на заранее неизвестное количество подзадач, объекты класса *Expert* могут созда-

ваться и уничтожаться по мере необходимости, объект класса *Env* (среда) может изменять свое состояние за счет изменения значений атрибутов.

Например, i-ый объект в момент времени *t* можно описать в виде следующего кортежа:

$$O_t = \{C_i, S_i, P_{i(t)}, M_i, T_i, t\}, \qquad (8)$$

где C_i – класс объекта; S_i – множество атрибутов, характеризующих и однозначно идентифицирующих объект; $P_{i(t)}$ – множество атрибутов, отражающих состояние объекта в момент времени *t*; M_i – множество методов; T_i – множество временных характеристик объекта, определяющих фазы существования объекта; *t* – время.

Наличие в модели (8) компонента *M* (множество методов) дает возможность определить набор операций, которые задают правила манипулирования объектами ПрО. Так как все сущности ПрО представлены множествами, то такими операциями, в частности, могут быть: включение или исключение элемента из множества, определение принадлежности элемента множеству, изменение значения идентифицирующего атрибута и другие.

Следовательно, теоретико-множественное представление модели ПрО позволяет на уровне построения ОМ определить не только сущности ПрО, но и набор операций по ее модификации. Например:

- включение (исключение) задачи во множество подзадач;
- включение (исключение) эксперта во множество экспертов;
- другие.

Реализация этих операций позволит выполнять, при необходимости, адаптацию модели. Если это относится к изменению состава подзадач или экспертов, то можно говорить о структурной адаптации модели. Если существует необходимость в актуализации, обновлении, очистке или модификации знаний для решения подзадач, то это касается информационной адаптации.

Структурную адаптацию определим как модификацию модели в ответ на следующие возможные возмущения среды:

- декомпозиция подзадачи (в виду ее сложности);
- удаление подзадачи (в силу ее тривиальности);
- включение в экспертную группу новых исполнителей;
- исключение эксперта из группы.

Информационную адаптацию определим как модификацию модели в случае, если:

- изменяется содержательная постановка задачи S_i (текст задачи, спецификация требований, ограничения);
- изменяется состав экспертной группы для решения задачи S_i;
- знания Z_i для решения одной или нескольких задач отсутствуют;
- знания Z_i для решения одной или нескольких атомарных задач устаре-вают (теряют свою актуальность) и требуется их обновление.

Как следует из выражения (8), все элементы модели предметной области имеют характеристики, которые позволяют фиксировать состояние объекта модели в любой момент времени. Например, для объектов ПрО можно определить следующие фазы:

- активная фаза – объект входит в модель предметной области и участвует в информационном процессе;
- фаза ожидания – объект входит в модель, но не участвует в информационном процессе;
- архивная фаза – объект исключен из модели, но при этом сохраняется и доступна вся информация об объекте;
- фаза удаления – объект исключается из модели, вся информация о нем уничтожается.

Тогда (для исследуемого класса задач) фазы существования объектов *Task* и *Expert* можно интерпретировать следующим образом:

- задача инициирована, назначена эксперту, ожидается получение паттерна знаний от эксперта (активная фаза);
- задача инициирована, но не решена (фаза ожидания);
- задача решена, получен паттерн знаний, возможно использование знаний для решения новых задач (фаза архивации);
- задача «устарела», знания потеряли акту-

альность (фаза удаления);
- эксперт приглашен в состав экспертной группы, ожидает назначения задачи (фаза ожидания);
- эксперт формирует паттерн знаний для решения задачи (активная фаза);
- эксперт не участвует в решении текущих подзадач (фаза архивации);
- эксперт удаляется из состава экспертной группы (фаза удаления).

Нетрудно видеть, что такой подход к моделированию ПрО позволяет при необходимости менять состав элементов модели на протяжении всего жизненного цикла ПрО. Решение об адаптации ПрО принимается на основании анализа состояний (фаз) объектов. Введение допустимых операций, а также наличие механизма смены фаз позволяет говорить о возможности изменять границы модели предметной области, то есть адаптировать ее к изменениям внешней среды.

С учетом проведенных рассуждений предлагается методология моделирования ПрО, основанная на теоретико-множественном ее представлении. Общая схема методологии включает следующие основные этапы:

1. *Построение концептуальной модели modC*. Такую модель можно построить, например, с использованием методологии структурного моделирования «сущность-связь», взяв за основу компоненты обобщенной модели (6). Далее необходимо уточнить состав, типы и атрибуты объектов КМ, определить фазы их жизненного цикла.
2. *Построение онтологической модели modO*. Необходимо выполнить формализацию онтологии ПрО, то есть выбрать структурное и информационное представление составляющих ОМ, определить набор операций по модификации модели и правила их выполнения.
3. *Структурная и информационная адаптация ПрО*. Решение об адаптации принимается на основании анализа состояний (фаз) объектов ПрО. Для исследуемого класса ЗПР это объекты классов *Task* и *Expert*.
4. *Построение физической модели ПрО*. На этом этапе разрабатывается архитектура программной системы, которая позволяет реализовать структурное и информационное представление модели ПрО и обеспечивает возможность их модификации при адаптации ПрО к изменениям среды.

Таким образом, описанная методология покрывает важные этапы моделирования ПрО. Кроме того, она позволяет строить адаптивную модель, способную реагировать на изменения состояний внешней среды.

Структурная составляющая модели ПрО

Для практического использования предложенной методологии (при переходе от ОМ к ФМ) пердлагается выбрать способ формализации онтологий (4)-(7), используя аппарат теории графов. Это позволит построить граф-модель структуры ПрО в более технологичной форме, удобной для организации, хранения и алгоритмической обработки.

Очевидно, что реализация концептуальной модели заключается в построении связного ациклического графа (дерева), узлы которого определяют иерархию подзадач исходной задачи, а дуги – уровень их вложенности. Каждой вершине графа ставится в соответствие набор атрибутов *Attr*, значения которых однозначно идентифицируют задачу, эксперта и определяют фазы существования объектов ПрО:

$$Attr^{id} = < id, task, name, addr, status, inf >, \quad (9)$$

где: *id* – уникальный идентификатор вершины (задачи); *task* – постановка задачи (требования к решению); *status* – состояние задачи (например, *Created* – задача инициирована, *Assigned* – задача назначена эксперту, *InProgress* – задача находится в процессе решения, *Done* – задача решена, *Canceled* – задача «устарела»); *name* – уникальный идентификатор эксперта; *addr* – адрес эксперта; *inf* – информационная составляющая (фактически решение задачи, представленное в одном из допустимых форматов).

Построенное таким образом атри-

бутивное дерево имеет следующий вид (рис. 1).

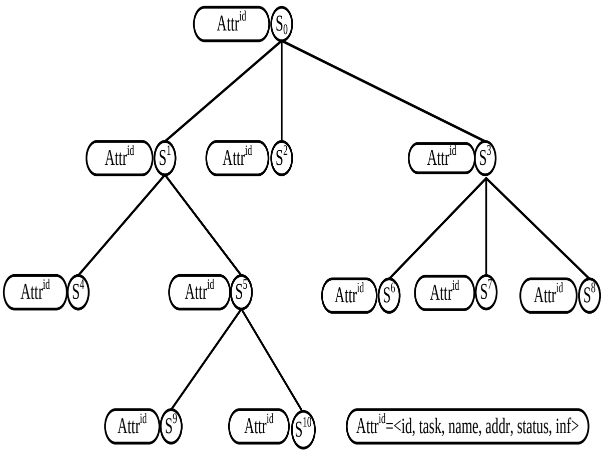


Рис. 1. Структурная модель ПрО

Предложенный способ формализации модели позволяет описывать структуру задачи с дальнейшим ее разбиением на необходимое количество подзадач.

Информационная составляющая модели ПрО

Построенная в виде атрибутивного дерева структурная модель ПрО не решает проблему представления знаний. В контексте ЗПР знания представляют решения подзадач целевой задачи. Для работы эксперта с определенным фрагментом предметной области, необходимо определить способ представления знаний (представления вершин дерева целевой задачи) и, как результат, построить информационную составляющую модели ПрО.

Согласно схеме распределенного решения сложных задач, каждый эксперт получает постановку подзадачи, реализовывает ее решение и полученный результат представляет в виде некоторого паттерна знаний. Такой паттерн можно рассматривать как структурированную информацию, необходимую и достаточную для решения задачи или ее части [8]. Предлагается строить два типа фреймов: для описания подзадач *P* и для описания паттернов знаний *Z*. Структура *P* определяет постановку задачи эксперту, структура *Z* описывает паттерн знаний для ее решения:

$$P = (Id, Text, Spec); Z = (Y, Tool), \quad (10)$$

где *Id* – идентификатор задачи; *Text* – текст постановки задачи; *Spec* – требования к решению; *Y* – решение (знания); *Tool* – средство реализации *Y*.

Слоты фрейма *P* заполняются центром, слоты фрейма Z на начальном этапе решения пусты.

Проведем модификацию концептуальной модели (6) путем добавления компонент, представляющих фреймы *P* и *Z*:

$$modC = (Entity, Relation, ERD \mid P, Z). \quad (11)$$

Первые три компонента модели (11) представляют структурную составляющую модели ПрО. Информационная составляющая представлена фреймами *P* и Z. Таким образом, модель (11) в рамках одной формальной структуры содержит компоненты, определяющие семантику задачи и знания для ее решения. Временной параметр *t*, а также механизм смены фаз позволяют отслеживать состояние модели и принимать решение о ее адаптации.

Адаптация модели осуществляется за счет операций включения (исключения) элементов и изменения значений идентифицирующих атрибутов объекта *Attr*. Реализация указанных операций требует уточнения модели (11) путем введения алгоритмической компоненты *Alg*:

$$modC = (Entity, Relation, ERD \mid P, Z \mid Alg). \quad (12)$$

Компонент *Alg* представляется набором алгоритмов, необходимых для выполнения операций по модификации граф-модели, что гарантирует возможность ее структурной адаптации.

Нетрудно заметить, что построенная выше модель (12) позволяют устранить статичность и неделимость модели П.Чена. Действительно, каждый терминальный узел, представленный фреймами *P* и *Z*, определяет вполне определенный фрагмент предметной области, который в дальнейшем может быть модифицирован экспертом (т.е. удовлетворяется свойство информационной адаптивности). Наличие алгоритмического компонента позволяет изменять состав множества объектов классов *Task* и *Expert,* что удовлетворяет свойство структурной адаптивности ПрО. Кроме того, изолирован-

ность частей модели позволяет реализовать их в отдельных программных компонентах, что обеспечивает формирование информационной части модели удаленными экспертами.

Построение физической модели ПрО

Физическая модель (ФМ) является результатом преобразования онтологической модели в информационно-программный компонент. В настоящее время для решения обозначенной проблемы существует большое количество инструментальных средств в рамках, так называемых, CASE-технологий. Анализ существующих CASE-технологий показал, что при построении ФМ значимый положительный эффект достигается в случае, когда предметные области решаемых задач хорошо структурированные и статичные (не требуют адаптации к изменениям среды), когда при разработке используются постоянные (локальные и распределенные) источники информации (как правило, реляционные БД), а сами разработчики и эксперты предметной области формируются по локальному принципу. Следует также заметить, что сложность, а порой и избыточность инструментов CASE-технологий, требует значительного участия в процессе моделирования инженеров знаний и программистов, и при этом не гарантирует актуальность и подтвержденную компетентность их знаний. Очевидно, что все это противоречит специфике современных ЗПР.

Для решения проблемы предлагается подход, основанный на использовании методологии и архитектурных принципов технологии распределенного моделирования High Level Architecture (HLA), регламентируемой стандартом IEEE 1516 [9]. Данная технология представляет собой совокупность методик, решений и стандартов для обеспечения взаимодействия территориально распределенных участников моделирования. Архитектура и стандарты HLA-тех-нологии ориентированы на создание распределенных динамических моделей, для которых характерно взаимодействие групп однотипных объектов, обменивающихся информацией о своем текущем состоянии. Отдельные участники модели, вне зависимости от их типа (программа, человек, аппаратное устройство), называются федератами. Совокупность федератов образует федерацию. Основная цель HLA – организация и поддержка информационного обмена между федератами. Взаимодействие федератов осуществляется инфраструктурой реального времени Run-Time Infrastructure (RTI).

Существенным является то, что создание модели с использованием сервисов стандарта HLA допускает возможность изменения состава участников, при котором новые федераты могут подключаться и прекращать работу непосредственно в процессе моделирования. Кроме того, стандарт HLA наследует концепции, которые позволяют описывать архитектуру и интерфейс системы, способной обеспечить взаимодействие нескольких географически удаленных друг от друга компонентов модели. Таким образом, отмеченные свойства позволяют использовать HLA в качестве основы для разработки технологии построения ФМ.

Технологию построения ФМ определим как процесс преобразования ОМ в модель программного компонента, реализованного в соответствии с архитектурными принципами стандарта HLA. Опишем общую схему построения ПрО задачи (1), включив в нее всех участников моделирования (2).

Основными участниками в данном случае являются центр, а также группа локальных и удаленных экспертов *E*. Центр формулирует задачу *Text(E)* экспертам и ожидает получение паттерна знаний *Inf(E)*. Совокупность паттернов образует ПрО и решение задачи *Solution*. Между экспертами и центром возможны различного рода коммуникации.

Опишем компоненты приведенной схемы в терминах HLA-архитектуры. Согласно теоретико-множественному представлению модели ПрО (8), каждый объект модели принадлежит к определенному классу: *Task, Center, Expert*. Так как объекты классов *Center* и *Env* существуют в единственном экземпляре, то ФМ можно представить в виде HLA-федерации, которая будет соответствовать объекту класса *Env* и управляться объектом класса *Center*.

Федерация формируется из объектов классов *Task* и *Expert*, каждый из которых представлен отдельным независимым федератом. Коммуникации между участниками сцены осуществляется средствами, предоставляемыми инфраструктурой RTI (рис. 2).

В соответствии с условиями задачи (1) и свойствами ПрО, количество объектов классов *Task* и *Expert* может изменяться. Эта возможность обеспечивается тем, что инфраструктура RTI поддерживает службы Join Federation Execution (создание нового федерата) и Resign Federation Execution (удаление федерата из федерации). Вызов этих служб позволит выполнить структурную модификацию ПрО.

Для адаптации ПрО можно выполнить декомпозицию отдельных подзадач и привлечь для их решения необходимое количество новых экспертов. На уровне программной реализации ФМ это достигается путем создания новых объектов-федератов и присоединения их к федерации.

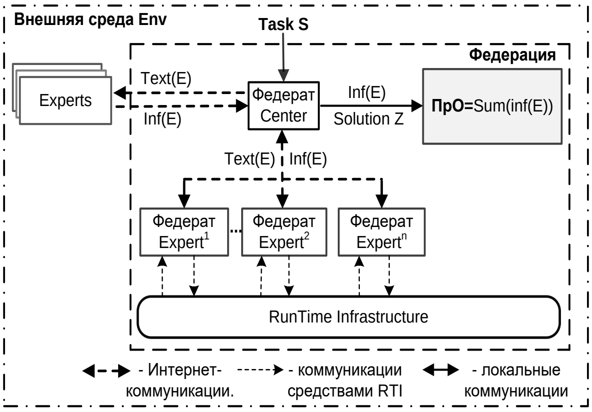


Рис. 2. ФМ в архитектуре HLA

Таким образом, на первом этапе необходимо построить укрупненный вариант ФМ, компоненты которой смоделированы в соответствии с концепциями HLA, и фактически предоставляют механизм управления динамическим процессом построения ПрО ЗПР.

Далее необходимо обеспечить соответствие между ФМ и структурной граф-моделью ПрО. К сожалению, в стандарте HLA отсутствует спецификация, описывающая схему иерархического подчинения федератов. Федераты позиционируются как независимые участники в составе федерации. Предлагается, не нарушая общей парадигмы HLA-архитектуры, модифицировать модель федерата путем введения атрибутов, идентифицирующих задачу и эксперта и дополнительного атрибута владения. В ситуации, когда эксперт E_i выполняет декомпозицию подзадачи S^i, то значение атрибута владения федерата E_i представляется списком уникальных идентификаторов его подзадач: $List=(id_1, id_2, \dots id_k)$. Такие же индексы присваиваются соответствующему атрибуту вновь созданных федератов.

Таким образом, такая модификация модели федерата гарантирует корректную интеграцию фрагментов знаний распределенных экспертов в соответствии с иерархией подзадач (рис. 3).

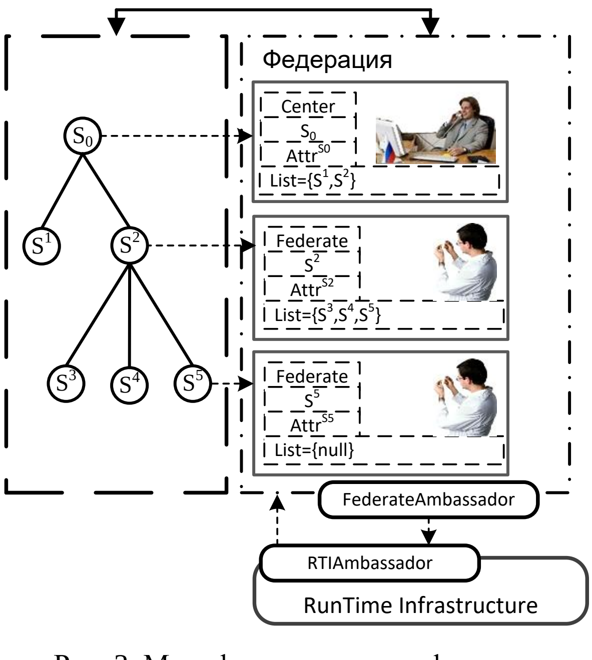


Рис. 3. Модификация модели федерата атрибутом владения

Завершающим этапом построения ФМ является формирование ПрО из полученных от экспертов фрагментов знаний *Z*.

На протяжении всего жизненного цикла федерации федерат-центр следит за процессом поступления паттернов знаний от распределенных экспертов. Как только этот процесс завершается, необходимо перехо-

дить к следующей фазе – интеграции фрагментов знаний в рамках единой ПрО. То есть, на уровне ФМ необходимо иметь механизм определения степени готовности ПрО для запуска процесса интеграции фрагментов. Для этого можно использовать паттерн «Publisher/ Subscriber» (издатель-подписчик), который в архитектуре HLA реализован средствами RTI для обмена данными между центром и федератами.

С учетом проведенных рассуждений общую схему технологии построения ФМ ПрО можно описать в виде следующих основных этапов:

1. *Построение начальной федерации (укрупненный вариант ФМ).* Такую модель можно построить, включив в состав федерации все множество объектов онтологической модели в виде федератов соответствующих классов.
2. *Уточнение структуры федератов*. Это реализуется путем добавления идентифицирующих атрибутов федератов и атрибута владения. Для этого можно, например, выполнить парсинг структурной составляющей модели ПрО, представленной в одном из допустимых форматов представления графов.
3. *Определение степени готовности модели ПрО.* Необходимо выполнить опрос значения коэффициента автономности федератов. Это можно реализовать, например, средствами RTI для обмена данными внутри федерации.
4. *Интеграция фрагментов знаний в единую структуру.* Решение принимается на основании анализа степени готовности модели ПрО и реализуется путем проверки значения коэффициента автономности федерата-центра.

Таким образом, описанная выше технология позволяет построить ФМ ПрО, соответствующую ее ОМ. Физическая модель фактически представлена в виде архитектуры программной системы, компоненты которой обеспечивает возможность адаптации ПрО за счет динамического изменения состава участников федерации.

Заключение

В статье рассматривается проблема моделирования ПрО для адаптивных СППР, которые функционируют в условиях быстро изменяющейся среды. Рассмотрены существующие методологии моделирования ПрО. Показано, что их прямое использование не позволяет устранить статичность модели и удовлетворить свойство адаптивности ПрО. Предложена методология построения адаптивной модели ПрО, основанная на традиционной трехэтапной схеме моделирования ПрО (КМ→ОМ→ФМ). Суть методологии заключается в построении КМ ПрО с использованием структурной методологии «сущность-связь» и уточнении ее до уровня ОМ с использованием теоретико-множественного представления. Выделены соответствующие этапы методологии. Предложено структурно-информационное представление модели ПрО ЗПР, введен набор операций на модели, позволяющий выполнить структурную и информационную адаптацию ПрО. На основе стандарта распределенного моделирования HLA разработана технология построения ФМ в форме федерации, сервисы которой реализуют механизмы динамического изменения состава элементов модели, что удовлетворяет основному требованию адаптивности ПрО.

Литература

1. Ткачук, Е.О. (2002). *Требования к адаптивным системам поддержки принятия управленческих решений.* Известия ТРТУ, 2, 248–251.
2. Башина, О.Э., Дик, В.В., Уринцов, А.И. (2014). *Онтология взглядов на этапы развития систем поддержки принятия решений в рамках перехода к информационному обществу.* Статистика и Экономика, 1, 174-183. doi: 10.21686/2500-3925-2014-1-174-183
3. Вальвачев, А.Н., Виссия, Х., Краснопрошин, В.В. (2008). *Технология выполнения IT-проектов коллективами распределенных исполнителей. Искусственный интеллект,* 3, 63–69.
4. Черемных, С.В. (2001). *Структурный анализ систем: IDEF-технологии.* М.: Финансы и статистика.
5. Чен, П. П-Ш. (1995). *Модель "сущность-связь" - шаг к единому представлению данных.* СУБД, 3, 137-158.
6. Якобсон, А., Буч, Г., Рамбо, Дж. (2002). *Унифицированный процесс разработки программного обеспечения.* СПб.: Питер.

7. Лаврищева, Е.М. (2016) *Теория объектно-компонентного моделирования программных систем.* М.: ИСП РАН. Retrieved from http://www.ispras.ru
8. Виссия, Х., Краснопрошин, В.В., Вальвачев, А.Н. (2011). *Интеллектуализация принятия решений на основе предметных коллекций.* Вестник Белорусского государственного университета, Сер.1, 3, 84-90.
9. IEEE Std 1516-2010. (2010). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules.* Retrieved from IEEE Xplore Digital Library.

References

1. Tkachuk, E.O. (2002). *Trebovaniya k adaptivnym sistemam podderzhki prinyatiya upravlencheskikh reshenii.* Izvestiya TRTU, 2, 248–251.
2. Bashina, O.E., Dik, V.V., Urintsov, A.I. (2014). *Ontologiya vzglyadov na etapy razvitiya sistem podderzhki prinyatiya reshenii v ramkakh perekhoda k informatsionnomu obshchestvu.* Statistika i Ekonomika, 1, 174-183. doi: 10.21686/2500-3925-2014-1-174-183
3. Val'vachev, A.N., Vissia, H., Krasnoproshin, V.V. (2008). *Tekhnologiya vypolneniya IT-proektov kollektivami raspredelennykh ispolnitelei.* Iskusstvennyi intellekt, 3, 63–69.
4. Cheremnykh, S.V. (2001). *Strukturnyi analiz sistem: IDEF-tekhnologii.* M.: Finansy i statistika.
5. Chen, P. (1995). *Model' "sushchnost'-svyaz'" - shag k edinomu predstavleniyu dannykh.* SUBD, 3, 137-158.
6. Jacobson, A., Buch, G., Rambo, J. (2002). *Unifitsirovannyi protsess razrabotki programmnogo obespecheniya.* SPb.: Piter.
7. Lavrishcheva, E.M. (2016) *Teoriya ob"ektno-komponentnogo modelirovaniya programmnykh sistem.* M.: ISP RAN. Retrieved from http://www.ispras.ru
8. Vissia, H., Krasnoproshin, V.V., Val'vachev, A.N. (2011). *Intellektualizatsiya prinyatiya reshenii na osnove predmetnykh kollektsii.* Vestnik Belorusskogo gosudarstvennogo universiteta, Ser.1, 3, 84-90.
9. IEEE Std 1516-2010. (2010). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules.* Retrieved from IEEE Xplore Digital Library.

RESUME

H.V. Karkanitsa, V.V. Krasnoproshin

Subject Domains Modeling for Adaptive Decision Support Systems

Any decision-making problem is being solved under the conditions of some environment. Currently the decision-making environment is changing rapidly. On the one hand, the intensity of new knowledge appearance is significantly increases. On the other hand, the network content and knowledge of some subject areas quickly become outdate. Under such circumstances, the risks of using unreliable, as well as outdated knowledge for decision-making, are increased. One of the ways to eliminate these risks is to involve qualified experts. In the process of solving the decision-making problem (due to the variability of the environment), a constant refinement of the subject domain is required. In such a situation, an actual problem is the development of an Adaptive Decision Support Systems (ADSS), capable of responding to changing decision-making environment. The development of ADSS requires the construction of an adaptive model of the subject domain and the development of corresponding algorithms for its modification.

In this study, a methodology for constructing an adaptive subject domain model for problems that are solved on the base of distributed expert knowledge is presented. Existing methodologies for subject domain modeling are considered. It is shown that their direct use does not allow to eliminate the static nature of the model and to satisfy the property of adaptability of the subject domain. A structural and information representation of the domain model is proposed. A set of operations which makes it possible to perform structural and information adaptation of the subject domain is introduced. A technology for constructing a functional model of subject domain in the form of federation has been developed on the basis of High Level Architecture standard.

Надійшла до редакції 25.09.2018